How to setup

Project contains the following files:

Public Folder : contains css and javascript files for the front-end

- abc.css
  for user main page
- home.js
  for login and register page
- bcd.js
  for user main page
- trade.js
  for approve or reject trade request page

views folder: contain pug files for html rendering

- friendsProfile.pug
  page contains all cards of friend
- home.pug
  page contains login/register
- tokenDetain.pug
  page contain details of cards
- trade.pug
  page contains trade approval
- userMain.pug
  main page for all user data

other files:

- **index.js**
  index.js is the entry point of the project contains all the routes
- card.js
  perform all the operations related to card schema like getting cards
- package.json
  contains project info
- user.js
  contains all the functionalities related of user like register/login, authenticating, send delete and approve friends requests, send delete and approve trade requests

How to start:

- Open Connection of mongodb server
- Open command prompt and change folder to project folder
- Run command : npm install (to download node module folder)
- To start server run command:
    - node index.js

Database models

Database a5 contains:

Cards schema as provided in database-initializer.js

1._id: the unique ID created by MongoDB when the card was inserted

2. artist: a string representing the name of the artist(s) that created the card art

3. name: a string representing the name of the card

4. cardClass: a string representing the class of player that can use the card

5. rarity: a string representing the rarity of the card

6. attack: an integer representing the attack value of the card

7. health: an integer representing the health value of the card

Users schema made in users.js contains all the user info:

```
 username: {
        type: String,
        unique: true,
        required: true
    },
    password: String,
    friends: Array,
    friends_requests: Array,
    cards: Array,
    trade_requests: []
}
```

Trade schema contains all the trade requests:

```
   sender: {
        username: String,
        cards: Array
    },
    receiver: {
        username: String,
        cards: Array
    }
```

Routes info:

Localhost:3000 is used for port

Api's	**info**	**Success return**	**Failure return**
Localhost:3000/	For returning login page	Return login page	error
`/userhome/:token`	For returning user main page	Return page contain all user info	error
`/cards/:name`	For returning card detail page	Return card details page	error
`/friends/:token`	For getting possible friends data	Return possible friends page	error
`/specific/:token`	Return user data	Return user data	error
`/friend/:name`	Returns friend profile page	Return friends profile page	error
`/trade/:tid`	Returns trade profile page	Returns trade profile page	Error
`/register`	Register user in database	Return true if registered	Return false if not registeed
`/login`	Check login detail valid or not	If valid true	If not valid return false
`/frndrequest/:token`	Send friend request	Add request to receiver data	error
`/apprequest/:token`	Approve a friend request	Update both sender and receiver friends list	error
`/delFrequest/:token`	Delete a friend request	Remove request from receiver list	error
`/apTRequest`	Approve the trade request	Update cards on both sender and receiver data and delete request	error

`/sndTRequest/:token`	Send trade request	Add request to receiver data Nd trade schema	error
`/delTRequest`	Delete trade request	Delete request from receiver data and trade schema	error